Korean Patent No. 2003-0036787

Job No.: O-03281
Ref.: KR2003-0036787/PU030107/JDH(SUSAN)/ORDER NO. ART620
Translated from Korean by the McElroy Translation Company
800-531-9977 customerservice@mcelroytranslation.com

Laid-Open Patent Gazette No. 10-2003-0036787 (5/9/2003). Part 1.

Patent 2003-0036787

(19) Korea Intellectual Property Office (KR)
(12) Laid-Open Patent Gazette (A)

(51) Int. Cl.[7] H 04 L 12/22
(11) Laid-Open No.: Patent 2003-0036787
(24) Laid-Open Date: May 9, 2003

(21) Filing No.: 10-2003-7003776

(22) Filing Date: March 14, 2003
Translation Submission Date: March 14, 2003
(86) International Application No.: PCT/US2001/28605 (87) International Publication No.: WO 2002/23797
(86) Filing Date of International Application: September 14, 2001 (87) International Publication Date: March 21, 2002

(81) Designated States: Domestic Patents: United Arab Emirates, Antigua and Barbuda, Albania, Armenia, Austria, Australia, Azerbaijan, Bosnia-Herzegovina, Barbados, Bulgaria, Brazil, Belarus, Belize, Canada, Switzerland, China, Costa Rica, Cuba, Czech Republic, Germany, Denmark, Dominican Republic, Algeria, Ecuador, Estonia, Spain, Finland, United Kingdom, Georgia, Hungary, Israel, Iceland, Japan, Kenya, Kyrgyzstan, North Korea, South Korea, Kazakhstan, St. Lucia, Sri Lanka, Liberia, Lithuania, Luxembourg, Latvia, Morocco, Moldova, Madagascar, Macedonia, Mongolia, Malawi, Mexico, Mozambique, Norway, New Zealand, Poland, Portugal, Rumania, Russia, Sudan, Sweden, Singapore, Slovenia, Slovakia, Tajikistan, Turkmenistan, Turkey, Trinidad and Tobago, Tanzania, Ukraine, Uganda, Vietnam, South Africa, Colombia, Grenada, Ghana, Gambia, Indonesia, India, Yugoslavia, Zimbabwe Croatia, Sierra Leone,

AP ARIPO patents: Ghana, Gambia, Kenya, Lesotho, Malawi, Sudan, Sierra Leone, Swaziland, Uganda, Zimbabwe, Mozambique, Tanzania,

EA Eurasian patents: Armenia, Azerbaijan, Belarus, Kyrgyzstan, Kazakhstan, Moldova, Russia, Tajikistan, Turkmenistan,

EP European patents: Austria, Belgium, Switzerland, Cyprus, Germany, Denmark, Spain, Finland, France, United Kingdom, Greece, Ireland, Italy, Luxembourg, Monaco, Netherlands, Portugal, Sweden, Turkey,

OA OAPI patents: Burkina Faso, Benin, Central African Republic, Congo, Cote d'Ivoire, Cameroon, Gabon, Guinea, Equatorial Guinea, Guinea-Bissau, Mali, Mauritania, Niger, Senegal, Chad, Togo,

(30) Priority Claims: 60/232,599 September 14, 2000 (US)
69/233,054 September 15, 2000 (US)

(71) Applicant: Probix, Inc.
Bldg. 8 Suite A 200, 883 North Shoreline Boulevard, Mountain View, California, USA 94043

(72) Inventors:
David A. Lordemann
1724 Oak Avenue, Los Altos, California, USA 94024
Daniel J. Robinson
1992 Hasting Court, Santa Clara, California, USA 95051

Paul O. Scheibe
3 Stillcreek Road, Woodside, California, USA 94062

(74) Agent: Yeong-Hee Kim

Claims for Examination: Not filed

(54) SYSTEM FOR ESTABLISHING AN AUDIT TRAIL TO PROVIDE SECURITY FOR OBJECTS DISTRIBUTED VIA A NETWORK

(57) Abstract

A log file setup system and method are provided for use in generating an audit trail. A security server (18) maintains a log file of action conducted by the requester (10) and the security server in relation to the secured object (16). The object control, instantiated with the object (16) at the requester device (10), transmits an encrypted descriptor of the action to the security server (18), and if there is no safe connection to the security server (18), the requester device (10) is prevented from engaging in any action (viewing, editing, printing, etc.) whatsoever. The security server (18), in addition to recording the descriptor of the action conducted by the security server (18) in relation to the security of the object (16), also records information received from the requester device (10) in the log file, along with other data.

Representative Drawing

Figure 1

Technical field

The present invention relates to the establishment of an audit trail for the security of an object, such as a code, document, or image, distributed via a network.

Prior art

Contractors and the partners and clients of growing businesses generally use the Internet in order to seek out information, exchange code, documents, images, etc., in the course of business. With the increasing amount of business taking place over the Internet, interest has grown in protecting information exchanged or stored over the Internet from hackers, and hackers have been able to gain unauthorized access to such information, and have used it for their own financial advantage or have harmed information or systems in which such information is stored. In view of the enormous quantity of business transacted over the Internet, and the enormous value of this business, the safety of the objects stored and exchanged (including code, documents, images and anything represented in a digital format), and of the intellectual property contained in these objects, is paramount. In other words, it must not be possible for an individual or company to gain unauthorized access to said objects or to the intellectual property contained with them, it must not be possible to print them without permission, and must not be possible to edit without permission having been provided by the owner.

The security technology for objects and object exchange contains many components. One of these, the certificate, is a process of confirming the identity of the information requester or sender. Said certificate generally is formed by the use of a password. The drawback of this method is that the password can be lost, revealed, or stolen.

In stricter authentication processes, a digital certificate is used that is provided by an authentication agency. A digital certificate includes the owner's name, serial number, expiration date, and digital signature of the issuing agency (confirmation of sender and message data using public-key encryption, and data added to the certified message). Said certificate also includes the public key of the certificate owner. In the public-key encryption widely used in authentication, an individual has a private key and a public key generated by the certifying agency at the same time using an algorithm such as RSA. Said public key is published in one or more directories that include the certificate, while said private key is kept secret. The message is encrypted using the recipient's public key, and this is captured by the sender in the directory, and decrypted by the recipient's private key. The sender can encrypt a message with the sender's private key for message authentication; the recipient can verify the sender's identity by decoding the signature with the sender's public key.

Authentication is determined by whether the user has privileges (view, modify, etc.) with respect to the resource. For example, the system administrator can decide whether the user may access the system, and can decide what rights each user will have within the system, such as access to certain files, storage capacity, etc. Ordinarily, rights assignment takes place after authentication. Thus, if the user requests access to an object, the system confirms or certifies the user's identity first, then determines whether the user has access rights to the object, and how the user will use said object.

For object security, encryption may also be used. The plain text of the message is converted to ciphertext by encryption. In order to translate the encrypted object, the recipient must obtain an exact decryption key (for example, refer to the description of the above-described basic public-key architecture and public-key encryption). Generally, the cipher used in encrypting an object can be broken, but the more complex the encryption, the more difficult it is to break the cipher without the decryption key. A strong encryption system has a sufficiently wide range of possible keys to make the encryption unbreakable by trying all possible keys. In addition, a strong encryption system is not affected by known methods of code hacking, and will appear random to all standard statistical tests.

Other types of security methods can be used at computer locations in order to secure the total computer system. For example, many companies install firewalls to prevent access by unauthorized users to the company's data or programs. However, firewalls can be damaged, and do not guarantee the safety of the computer system against invasion. Another problem is that a

firewall does not prevent systems or system resources from being damaged by an unauthorized user located behind the firewall.

Message transmission can be made secret. Ordinarily, Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols are used to provide encrypted communication between server and client. These two protocols are integrated into the majority of web browsers and servers.

By implementing accountability, i.e., by tracking user actions that are either related to an object (such as an object request) or actually performed on an object (viewing, editing, printing, etc.), audit trails also provide security. Audit trails must be safe against unauthorized modification, for example it cannot be allowed for an unauthorized user to remove the evidence of his or her own action from an audit log. Because the auditing of requests and actions generates a large quantity of information, the system that generates the audit log must have the capability to store and efficiently process said information.

The above-described security apparatus can be used separately, but it is more common for several types to be used together. In addition to such ordinary apparatus, other security methods also exist in the prior art.

InterTrust Technologies Corporation has acquired numerous patents related to digital rights management. By means of InterTrust's Digibox container technology, information (including content and rules related to access to the content) is encrypted and can be stored in a Digibox container, which is essentially a software container. Said encryption key and container are transmitted from node to node in the virtual distribution environment (VDE). Said virtual distribution environment includes dedicated hardware or software, or a combination of these. Only a device integrated with the VDE that runs the appropriate InterTrust software can view the information within said container. An audit trail is generated and stored within said VDE, and can be shown.

An invention is needed that will secure objects (including code, documents, images, and software programs; basically, anything that can be represented in digital form) that are available on the Internet without authorized requesters running specific software on their computers in order to access secured information; a secure audit trail is also needed, in order to ensure accountability and non-refutability.

It is preferable that security duties, including storage of audit trails, be assigned to a third party, in order to reduce the processing and hardware load (including sufficient memory for storing enormous audit trails) of providing safety for the object servers. Finally, it is preferable that in order to demonstrate the integrity and non-refutability of the audit trail and provide wide-ranging security, information such as a description of the object secured by the audit trail, and

the security policy of the requested object, nonce of the requested object, serialization of the requested object, rights assignment, authentication and request.

An invention is needed that will secure objects (including code, documents, images, and software programs; basically, anything that can be represented in digital form) that are available on the Internet without authorized requesters running specific software on their computers in order to access secured information. For example, due to their limited budgets, even if students have their own computers, they can of course scarcely be expected to purchase software enabling the download of information like lecture notes and schedules (schools gradually are allowing the use of such information by authorized users). Additional preferable features of a digital rights management system include passing most security "duties" to a third party in order to reduce the object server's processing load from providing safety and providing single-use encryption keys that are safely transmitted between the requester and the "security server" instead of transmitting the encryption keys together with the encrypted data. Even after the object has been sent to the requester, it is preferable that the digital contents rights management system continue to provide security for the object.

Detailed explanation

The present invention provides a method and system for securing an object (anything represented in digital form, including code, documents, images, and software programs) distributed via a network. "Security" is the restriction of certain recipients from engaging in certain actions (such as viewing, printing, editing, copying) with respect to an object.

An object server that includes all secured objects and unsecured objects is furnished with software that regulates whether an object must be secured and what the security policy is (the type and level of security the object receives). Said security policy includes not only action policies related to actions such as whether the object can be printed or whether it can be edited, but also restrictions on the people who can view the object, the lifespan of the object, and the number of times the object can be shown. Object controls are the mechanisms that implement the security policy.

When the object server receives a request for an object, the software checks whether said requested object is secured. If the object has not been secured, the server sends said object to the requester. If the object has been secured, the software generates a new object including the serialization of the requested object, the security policy and description, as well as the time and certificate of the original request. Said new object is sent as a response to the requesting browser, along with a redirection command that causes the requesting browser to designate a "security server."

The security server, furnished with software that provides the security service, first receives and certifies the redirected request, and then acquires said requested object from its own unique cache or a server that includes the object, via secure transmission. Next, the security server encrypts said requested object using strong and non-malleable encryption, and combines it with mobile code (software that is, without explicit installation or execution by the recipient, sent from a remote system, transferred via a network, and downloaded and executed on the local system), the security policy, and object controls. The resulting package is returned to the requesting computer as a response to said redirected request.

Said requesting computer then attempts to run said mobile code in order to provide the requested object. Said mobile code runs a test in order to verify the instantiation of the object controls, and if these controls have been properly instantiated, when the request authentication has been satisfied, the requester requests the decryption key, which is sent to the requester by secure transmission. Said decryption key is a single-use key used only to decrypt this specific object. If said mobile code is executed successfully and the decryption key is acquired, said requested object is restricted by the security policy and object control.

A descriptor of the action of the requestor in relation to the object, and specific action related to the security server, is recorded in the log file, which can be used in inspection by authorized individuals such as the content owner or system administrator of the security server. Using said log file, an audit trail can be drawn up that describes specific actions carried out by the requester, what type of security policy was appropriate for each object, whether the object was transmitted, who requested which object, as well as acquired information, such as the number of times the object was accessed and the time the object was accessed.

Said security server is used to execute the majority of actions related to securing and transmitting requested objects. Accordingly, the object server is dedicated to the processing of requests for information, rather than expending processing resources on security issues. In addition, because the system administrator of the server deals with all time settings and administration of the object server, resulting savings are realized by the owner of the object server.

The method and system of the present invention differ from other object security methods and systems in that there is no need to install shared software on all computers involved in requesting objects and providing the requested objects.

In addition, the key used in encrypting and decrypting the object is a single-use key, and is not transmitted along with the encrypted object.

Brief description of the drawings

Figure 1 is a block diagram of the configuration elements of the object security system of the present invention

Figure 2a is a flowchart depicting the method of the present invention by which an object is secured.

Figure 2b is a flowchart depicting the method of the present invention by which an object is secured.

Figure 3a is a flowchart depicting the method of the present invention by which a log file is generated of the requester's actions with respect to a secured object.

Figure 3b is a flowchart depicting the method of the present invention by which a log file of object server action is generated.

Embodiments

Referring to Figure 1, connected to a network (in this embodiment, the Internet (20)), there are: a requester device (10) (a computer in this embodiment, but including any device that can act as a client in a client/server relationship); an object server (12) that contains an object (16) and security software (14) that indicates objects that must be secured; and a security server (18) that contains software (94) for providing security services. Objects (16) include code, documents, images, software programs, and anything else that can be represented in digital form. An attacker (22) is also present, corresponding to a person or device such as a computer or recorder that is used in order to gain unauthorized access to the secured object. Although one requester device (10), object server (12), and security server (18) are described here, it is possible for the method and system of the present invention to accommodate a plurality of requester devices (10), object servers (12), and security servers (18).

In the present embodiment, said object server (12) and security server (18) are Hypertext Transfer Protocol (HTTP) servers. Said requester device (10) must run a software program that operates as a World Wide Web browser (24). The request concerning the object (16) from the requester device (10) is relayed to the object server (12) by the browser (24) via an HTTP request. Likewise, the response to the request also follows the HTTP protocol.

As described above, the object server (12) runs the security software (14), and in the present embodiment the security software (140) [sic; (14)] is an extension of the HTTP server software. An authorized system administrator uses said security software (14) to designate which objects (16) have not been secured and which will be secured. If an object (16) is designated as having been secured, said security software (14) induces the administrator to designate the type and level (for example, the security policy) of security with respect to the object (16). Said security policy includes restrictions on the number of times the object may be viewed (cardinal

restrictions), object lifetime (temporal restrictions), and persons who can view the object, as well as action policies related to whether the object can be printed, edited, etc. The actions that the requester can perform on the object differ depending on the requester's identity. The object controls are the mechanisms that implement the security policy.

The security server (18) also runs the software (94) that is an extension of the HTTP server software. This software (94) provides security services for the object.

In Figure 2a, a requester requests an object (step 26). The object server that stores said requested object receives said request (step 28). If said object server has an independent authentication policy, the object server will carry out that policy, and will certify the request upon receiving it. Said security software inspects the HTTP request and determines (step 30) whether that request relates to a secured object. If the requested object has not been secured, the requested object is sent to the requester (step 32).

However, if the object is secured (step 30), said security software generates a secured request included in the response to the request of said security software (step 34) which is then re-sent to the security server. Said secured request is an object that includes the original request time and encryption data including authentication, in addition to the description, security policy, nonce, and serialization (verifying that only one approved version of the object can be used) of the requested object. Information related to authentication is governed by whether the object server has an independent authentication policy. If there is an authentication policy, said secured request will include the results of authentication. If there is not an authentication policy, the information will include said secured request.

Diverse services are provided by encryption. Said authentication can not only support authentication and the assignment of rights to a request, but can also protect the integrity of a file (for example, preventing unauthorized modification). Here it is possible also to protect the requester's individual privacy by using encryption. Another function of encryption is the prevention of repudiation (non-repudiation) and the detection of changes. A protocol is used that supports strong and non-malleable encryption. The protocol determines the type of encryption used, and whether exchange between the requester and server is necessary prior to encryption taking place (for example, it is often necessary to exchange the key so that the recipient can decrypt an object encrypted by the server).

Said enhanced request is included in the response to the requester, along with the command to re-send the request to the security server. Said re-sending must be transparent to the requester.

Said security server software decrypts said enhanced request (step 38). A shared key for encrypting and decrypting said enhanced request exists at both the object server and the security server. This key is generated when said software is installed on the object server.

Next, said security server software checks whether said enhanced request satisfies the requirements for a well-formed request (step 40). If the requirements for a well-formed request are not satisfied, the security server returns a message to the object server indicating an invalid request (step 42). Said object server sends a message concerning the invalid request to the requester. The system manager for said object server determines whether to send said message.

If said request is valid, the security server software authenticates said request (step 44).

The security server software compares the time and certificate in the re-sent request heading to those included in the enhanced request. If said security server software cannot authenticate the request (for example, a replay attack is indicated because the two request times are different from one another, or the requester identity in the re-sent request differs from the requester identity in the enhanced request), a message is returned to the object server indicating that authentication was not satisfied (step 46). If the request is authenticated, the security server software decrypts said request, and the requested object is acquired from the security server cache or from the object server (step 48). If there is a request, said security software transmits said object to the security server. If the security server must acquire the object from the object server, the object is sent via secure transmission.

When the security server has acquired the requested object, the security server software encrypts said object using strong encryption and non-malleable encryption and combines the object with mobile code (software sent from a remote system without explicit installation or execution by the recipient, transmitted via a network, and downloaded and executed on the local system), a security policy having authentication, included in the enhanced request, and object controls (step 50). The encryption of the secured request object acts to protect the object and the requester and provider of the object by ensuring the integrity, personal information, and authentication (if appropriate), as well as being a change-prevention and repudiation-prevention tool (so that a party to a transaction is unable to improperly deny a relationship to that transaction). Next, the result package is sent to the requester (step 52; see step B in Figure 2b).

In Figure 2b, the requester receives the response and attempts to execute said mobile code (step 54). If said mobile code is executed, the security policy and object controls for the requested object are instantiated on the requester's computer (step 54). Said mobile code executes a test to determine whether the object controls have been accurately instantiated. If they have been accurately instantiated, if the requester needs a decryption key (step 56), the requester requests the decryption key from the security server (step 58). The security server software authenticates said request (step 60). If it is not possible for said security server software to authenticate that request, a message concerning that result is sent to the object server (step 62). However, if the message is authenticated, said security server software returns the requested key to the requester by secure transmission (step 64), and the requested object is decrypted (step 66).

The key used by the security server in encrypting and decrypting the object is a single-use key. Said single-use key is provided either by a "seed" for randomly generating the key, determined at the time of installation of the security server software, or by another well-known means, most typically by certificate.

If said mobile code is executed, the requester can view the object under the restrictions imposed on the object by the object controls or security policy (step 68).

As shown in Figure 3a, the log file for action carried out on the object by the requester is maintained in order to establish an audit trail. Said log file can be used for inspection by the system administrator of the security server. Using said log file, an audit trail is created that describes what sort of security policy is appropriate for each object, whether the object has been sent, and who requested which object.

If the requester attempts an action related to the object (viewing, printing, editing the object, etc.), the object controls will determine whether a network connection has been established (step 82). If there is an open connection, an encrypted descriptor of the action is sent to the security server, which records said descriptor together with some of the other data in a log file (step 88). The other data recorded in the log file includes "local data," namely server-side data including the server's local time zone, identity, and time, and the requester's network IP address. Said information is sent to the security server and if a verification is sent to the requester (step 94), the action with respect to the object is permitted (step 90). For example, as described above, the requester can view the requested object only if the mobile code is successfully instantiated and the decryption key has been received from the security server. First, if the object is displayed on the requester's computer, a descriptor of said event, namely viewing of the object, is sent to the security server. If no verification is sent to the requester, the requester's request to perform an action on the object is rejected (step 92).

If no secure connection to the security server has been established, the object controls will attempt to establish such a connection to the security server (step 84). If said connection is established (step 86), an encrypted descriptor of the action is sent to the security server, and said security server records said descriptor and other above-described data in the log file (step 88). The action is then permitted on the object (step 90). However, if a connection cannot be established (step 86), the requester's request to perform the action on the object is rejected (step 92).

As shown in Figure 3b, the security server stores the descriptors of actions conducted with respect to the encrypted object in the log file. These actions include the response to the object request, the transmission of the object to the requester, the receipt of the request for the decryption key and the transmission of the decryption key to the requester. When the security server carries out an action (step 74), the system software determines whether the action is

related to the transmission of a secured object or is related to a request for a decryption key (step 76). If said action is not related to the transmission of the secured object or the request for an encryption key, nothing at all is recorded in the log file (step 80). However, if said action is related to the secured object or to the encryption key, the descriptor of the action is recorded in the log file along with the time, local data and requester network IP address (step 78). For example, if the security server receives a enhanced request for a secured object, said security server will store the enhanced request in said log file, and along with it will be stored at least the time, local data and requester network IP address. When the security server has transmitted a package including the object combined with the mobile code, the record of this action is recorded in said log file.

In another embodiment, the requester can take actions with respect to the object but is "untethered" (that is, not connected to the security server). If untethered action is permitted by the security policy, the requester's action is recorded on the requester device, and the requester sends it to the security server when a connection to the security server has been established. A control is configured so that if a connection to the network is not established within a set timeframe, access to the object is restricted.

In yet another embodiment, the descriptors of the security server actions can be encrypted prior to being recorded in the log file. This embodiment can be used when persons other than the system administrator are permitted access to said log file.

Claims

1. A security system that secures an object by providing a log file for requested action and action conducted on the object, which is distributed over a communications network, wherein said security system comprises:

a) an object server connected to the network that runs software programs that specify a security policy for objects that have been or will be secured;

b) a requester device that requests the object from said object server, and is connected to the network;

c) a security server that runs other software programs that provide security services to objects specified as having been secured by said software program, wherein the software providing said security service comprises:

i) a receiving means wherein redirected and enhanced requests for a requested object from the requester device are received, and further wherein said enhanced requests correspond to the initial request of said requester device, and are generated by said object server, and further wherein redirected and enhanced requests with respect to a requested object are received,

including encrypted data related to the time and certificate of the original request, as well as the requested object's description, security policy, nonce, and serialization;

ii) a means for acquiring said requested security object from an object server or cache wherein said requested security object is stored;

iii) a means for encrypting said requested security object;

iv) a means for combining said requested security object with a movement code, security policy and object control;

v) a transmission means for the result file, which transmits the result file to the requester device, wherein the object requested must be provided to said requester device by said requester device executing said movement code, and which depends on the security policy and object control located on the requester device when executing said movement code, in order for said requester to use and view the object;

vi) a means for confirming whether said object control is appropriately instantiated;

vii) a means for providing the decryption key to the requester based on whether said key request is satisfactorily certified;

viii) a means for storage in the security server, wherein information on an event is recorded in the log file, and such events are part of the group comprising:

A) requests for action with respect to a request-security object, initiated by the requester device;

B) action performed on said request-security object by the requester device;

C) action related to the security of the request-security object by said security server.

2. The security system cited in Claim 1, wherein said log file is used to generate an audit trail.

3. The security system cited in Claim 1, wherein said recorded information is the start of the event.

4. The security system cited in Claim 1, wherein said recorded information is local data.

5. The security system cited in Claim 1 wherein said recorded information is the network IP address of the requester device that initiated the event.

6. The security system cited in Claim 1, wherein the information recorded in said log file includes a descriptor of the event.

7. The security system cited in Claim 1, wherein the information recorded in said log file includes the request transmitted to the security server.

8. The security system cited in Claim 1, wherein the information transmitted by said requester device to said security server is encrypted by protocol.

9. The security system cited in Claim 8, wherein the protocol including encryption of information provides strong encryption.

10. The security system cited in Claim 8, wherein the protocol including encryption of information provides non-malleable encryption.

11. The security system cited in Claim 1, further comprising a means for setting up a connection between the recipient device and security server in order to record information about the requester-device-initiated request for action, wherein said connection is set up if there is no currently existing connection between said requester device and said security server.

12. The security system cited in Claim 11, further comprising a means for refusing the requested action with respect to a secured object if it is not possible to set up a connection between said requester device and said security server.

13. The security system cited in Claim 1, further comprising a means whereby if a connection has been set up between said requester device and said security server, the action with respect to the requested security object is recorded by the untethered requester device in a file on the requester device, and said file is transmitted to the security server.

Drawings

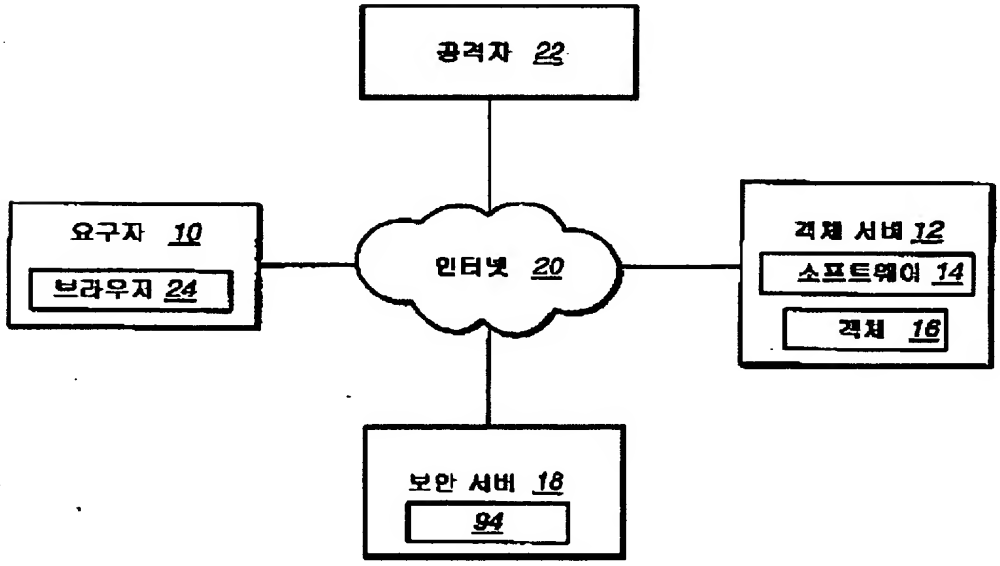


Figure 1

Key:
10 Requester
12 Object server
14 Software
16 Object
18 Security server

20 Internet
22 Attacker
24 Browser

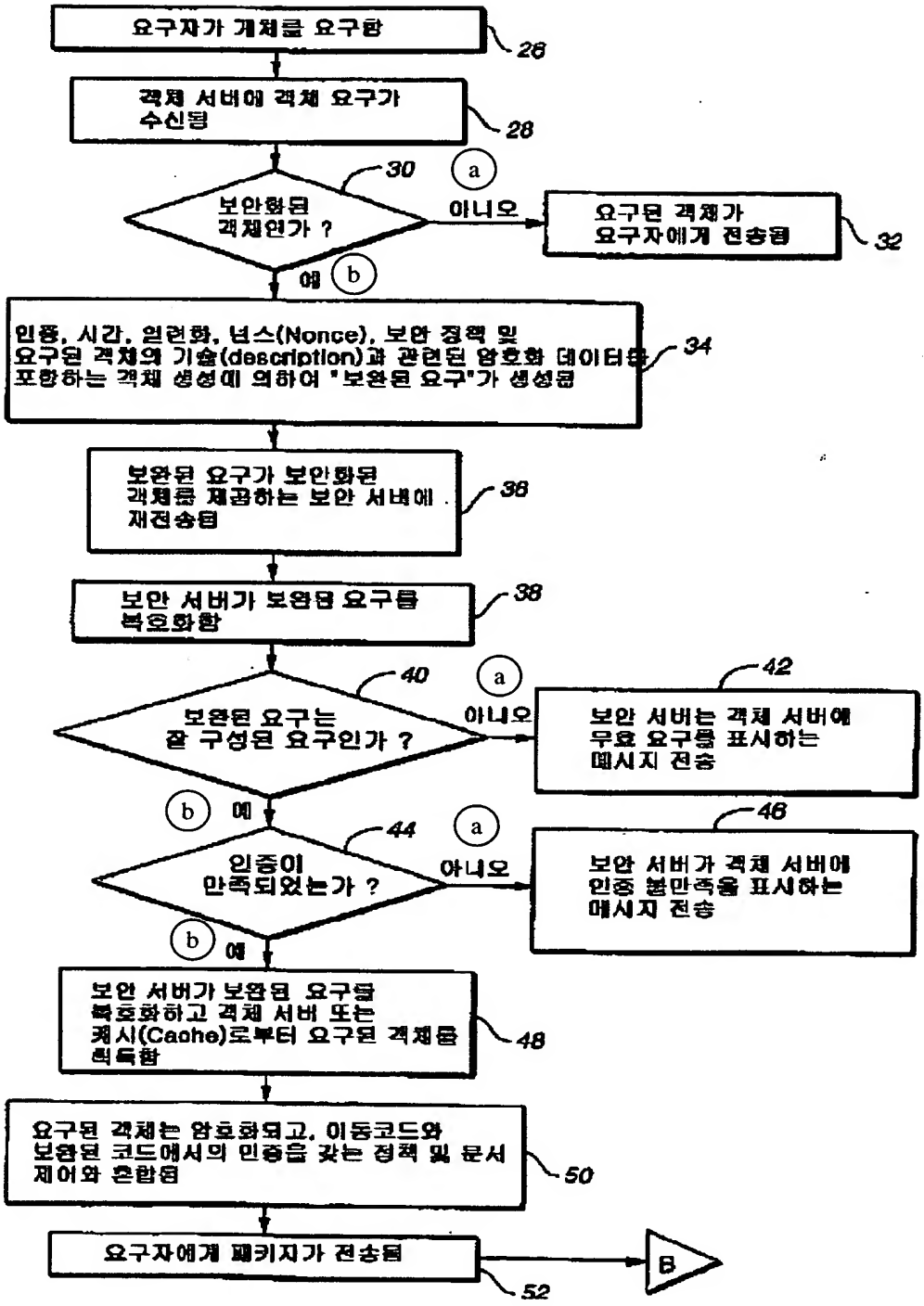


Figure 2a

Key: a No
b Yes
26 Requester requests object

28 Object request received at object server
30 Is it a secured object?
32 Requested object sent to requester
34 "Enhanced request" generated by the generation of an object containing encryption data related to certificate, time, serialization, nonce, security policy, and description of requested object
36 Enhanced request re-sent to security server that provides secured object
38 Security server decrypts enhanced request
40 Is the enhanced request well-formed?
42 Security server sends message to object server indicating invalid request
44 Has authentication been satisfied?
46 Security server sends message to object server indicating unsatisfactory authentication
48 Security server decrypts enhanced request and acquires requested object from object server or cache
50 Requested object is encrypted and combined with mobile code policy including authentication by enhanced code, and document controls
52 Package sent to requester

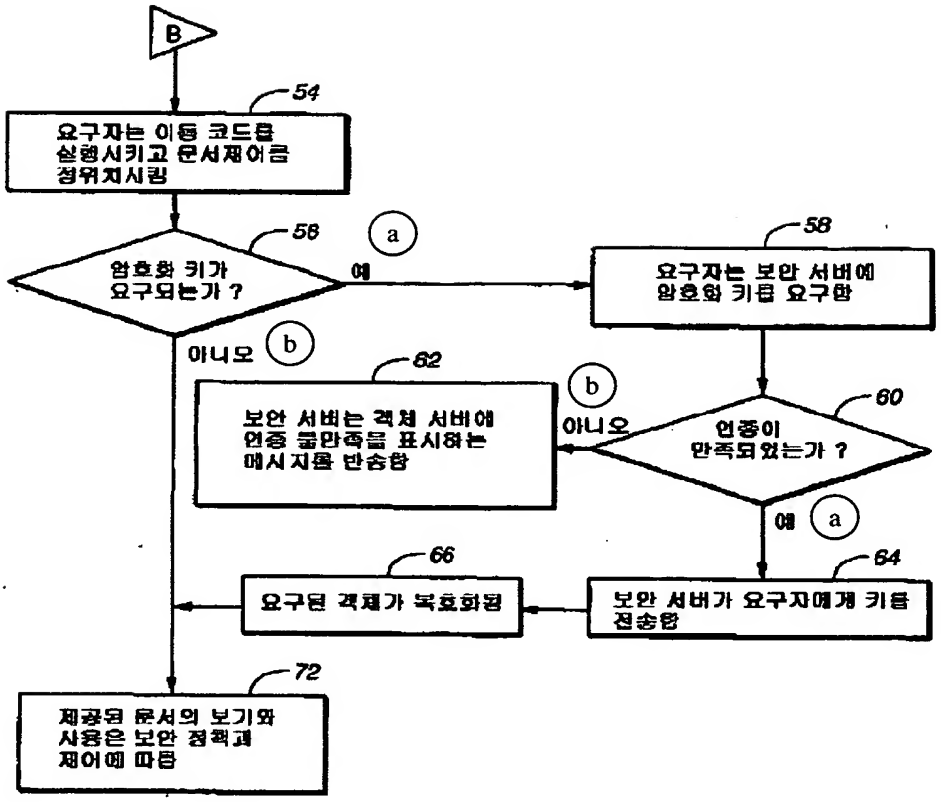


Figure 2b

Key:

a	Yes
b	No
54	Requester executes mobile code and puts document controls in position
56	Is an encryption key required?
58	Requester requests encryption key from security server
60	Has authentication been satisfied?
62	Security server returns message indicating unsatisfactory authentication to object server
66	Requested object is decrypted
64	Security server sends key to requester
72	Viewing and use of the provided document is governed by security policy and controls

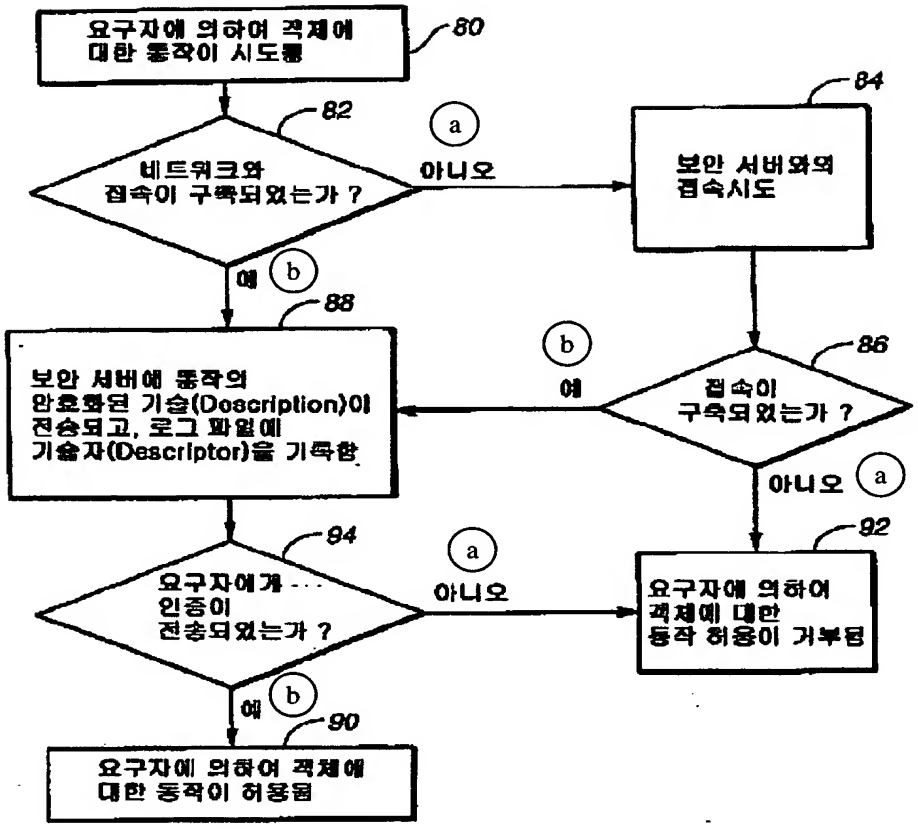


Figure 3a

Key:

a	No
b	Yes
80	Operation on object attempted by user
82	Has a network connection been established?
84	Attempt to connect to security server
86	Has a connection been established?
88	Encrypted description of the operation is sent to security server, and descriptor is recorded in log file
90	Permission for the operation by the requester on the object is granted
92	Permission for the operation by the requester on the object is denied
94	Has a certificate been sent to the requester?

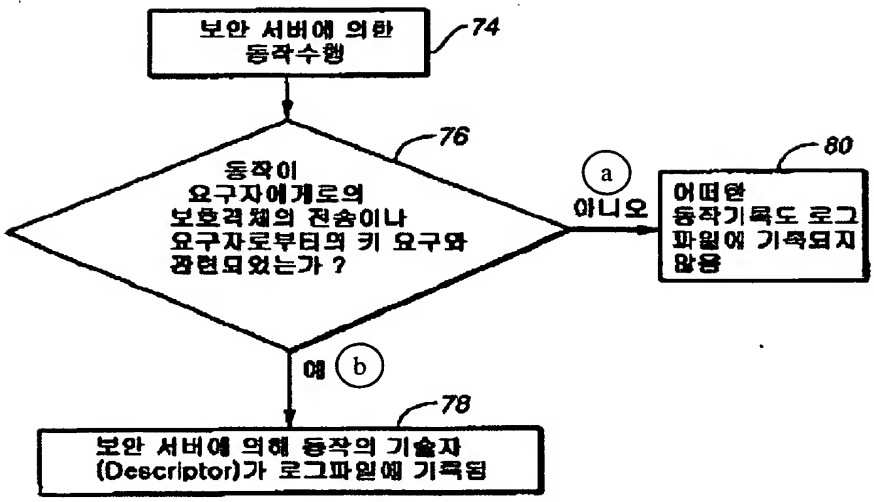


Figure 3b

Key:
- a — No
- b — Yes
- 74 — Operation performed by security server
- 76 — Is the operation related to the sending of a protected object or the request for a key from the requester?
- 78 — A descriptor of the operation is recorded in the log file by the security server
- 80 — No record of the operation is made in the log file

특2003-0036787

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.[7]
H04L 12/22

(11) 공개번호 특2003-0036787
(43) 공개일자 2003년05월09일

(21) 출원번호 10-2003-7003776
(22) 출원일자 2003년03월14일
번역문제출일자 2003년03월14일
(86) 국제출원번호 PCT/US2001/28605
(87) 국제공개번호 WO 2002/23797
(86) 국제출원출원일자 2001년09월14일
(87) 국제공개일자 2002년03월21일
(81) 지정국 국내특허 : 아랍에미리트 안티구아바부다 알바니아 아르메니아 오스트리아 오스트레일리아 아제르바이잔 보스니아-헤르체고비나 바베이도스 불가리아 브라질 벨라루스 벨리즈 캐나다 스위스 중국 코스타리카 쿠바 체코 독일 덴마크 도미니카연방 알제리 에쿠아도르 에스토니아 스페인 핀랜드 영국 그루지아 헝가리 이스라엘 아이슬란드 일본 케냐 키르기즈 북한 대한민국 카자흐스탄 세인트루시아 스리랑카 라이베리아 레소토 리투아니아 룩셈부르크 라트비아 모로코 몰도바 마다가스카르 마케도니아 몽고 말라위 멕시코 모잠비크 노르웨이 뉴질랜드 폴란드 포르투칼 루마니아 러시아 수단 스웨덴 싱가포르 슬로베니아 슬로바키아 타지키스탄 투르크메니스탄 터어키 트리니다드토바고 탄자니아 우크라이나 우간다 베트남 남아프리카 콜롬비아 그레나다 가나 감비아 인도네시아 인도 유고슬라비아 짐바브웨 크로아티아 시에라리온 AP ARIPO특허 : 가나 감비아 케냐 레소토 말라위 수단 시에라리온 스와질랜드 우간다 짐바브웨 모잠비크 탄자니아

EA 유라시아특허 : 아르메니아 아제르바이잔 벨라루스 키르기즈 카자흐스탄 몰도바 러시아 타지키스탄 투르크메니스탄

EP 유럽특허 : 오스트리아 벨기에 스위스 사이프러스 독일 덴마크 스페인 핀랜드 프랑스 영국 그리스 아일랜드 이탈리아 룩셈부르크 모나코 네덜란드 포르투칼 스웨덴 터어키

OA OAPI특허 : 부르키나파소 베넹 중앙아프리카 콩고 코트디브와르 카메룬 가봉 기네 적도기네 기네비쏘 말리 모리타니 니제르 세네갈 차드 토고

(30) 우선권주장 60/232,599 2000년09월14일 미국(US)
60/233,054 2000년09월15일 미국(US)

(71) 출원인 프로박스, 인크.
미국 캘리포니아주 94043 마운틴 뷰 노쓰 쇼어라인 불바드 883 빌딩 에이 스윗 에이200
(72) 발명자 로드만데이비드에이
미국캘리포니아주94024로스알토스오크애비뉴1724
로빈슨다니엘제이
미국캘리포니아주95051산타클라라해스팅코트1992
세이베풀오
미국캘리포니아주94062우드사이드스틸크리크로드3
(74) 대리인 김영희

심사청구 : 없음

(54) 네트워크를 통하여 분배되는 객체를 보안화하기 위한 감사추적 구축용 시스템

요약

감사 추적(audit trail)을 생성하는 데에 사용되는 로그 파일의 설정 시스템과 방법이 제공된다. 보안 서버(18)는 보안화된 객체(16)와 관련된 보안 서버 및 요구자(10)에 의하여 수행된 행위의 로그 파일을 유지한다. 요구자 장치(10)에서 객체(16)로 실체화된 객체 제어는 암호화된 행위의 기술자(description)를 보안 서버(18)에 전송하고, 보안 서버(18)와의 안전한 접속이 존재하지 않을 경우 요구자 장치(10)가 어떠한 행위(보기, 편집, 인쇄 등)도 취할 수 없도록 한다. 보안 서버(18)는 객체(16) 보안과 관련하여 수행된 보안 서버(18)의 행위에 대한 기술자를 기록하는 것 이외에도 요구자 장치(10)로부터 수신된 정보를 다른 데

이터와 함께 로그 파일에 기록한다.

대표도

도1

명세서

기술분야

본 발명은 네트워크를 통하여 분배되는 코드, 문서 및 이미지 등의 객체를 보안화하기 위한 감사 추적의 구축에 관한 것이다.

배경기술

협력 업체와, 유망 사업의 파트너 및 고객들은 사업의 과정에서 정보, 교환 코드, 문서 및 이미지 등을 찾기 위하여 인터넷을 일반적으로 사용하고 있다. 인터넷 상에서 이루어지는 사업이 증가됨에 따라, 인터넷 상에 저장되거나 교환되는 정보를 해커(hacker)로부터 보안화하는 것에 관한 관심도가 증가되었으며, 해커들은 이들 정보에 대한 불법 액세스가 가능하고, 그 정보를 자신들의 경제적 이득을 위하여 사용하거나 정보 또는 정보가 저장되는 시스템을 손상시킨다. 인터넷 상에서 이루어지는 막대한 양의 사업과 그 사업의 상당 가치를 고려하면, 저장 및 교환되는 객체(코드, 문서, 이미지 및 디지털 형식으로 표현된 모든 것을 포함함)와 그 객체 내에 포함된 지적 재산은 안전이 필수적이다. 즉, 상기 객체와 그 객체 내에 포함된 지적 재산은 권한 없는 개인이나 회사에 의해 액세스될 수 없어야 하고, 허가가 없는 한 인쇄될 수 없어야 하며, 소유자에 의하여 권한이 부여되지 않는 한 편집될 수 없어야 한다.

객체 및 객체 교환의 보안 기술은 많은 구성 요소를 포함한다. 이들 중 하나인 인증은 정보 요구자 또는 정보 송신자의 신원을 확인하는 프로세스이다. 상기 인증은 패스워드의 사용에 의하여 이루어지는 것이 일반적이다. 이 방법의 단점은 패스워드가 분실되거나 유출 또는 도난될 수 있다는 점이다.

보다 엄격한 인증 프로세스에서는 인증 기관에 의하여 제공된 디지털 인증서가 사용된다. 디지털 인증서에는 소유주의 성명과, 일련 번호와, 유효 기간 및 발행 기관의 디지털 서명(공개키 암호화를 사용하여 송신자와 메시지 데이터를 확인 및 인증하는 메시지에 부가되는 데이터)이 포함된다. 상기 인증서에는 인증서 소유주의 공개키도 포함된다. 인증 절차에서 광범위하게 사용되는 공개키 암호화에 있어서, 개인은 RSA와 같은 알고리즘을 사용하는 인증 기관에 의해 동시에 생성되는 공개키와 개인키를 갖는다. 상기 공개키는 인증서를 포함하는 하나 이상의 디렉토리(directory)에 공개되고, 상기 개인키는 비밀 상태로 유지된다. 메시지는 수신자의 공개키에 의하여 암호화되고, 이는 송신자에 의하여 디렉토리에 캡쳐되며, 수신자의 개인키에 의해 복호화된다. 메시지 인증을 위하여 송신자는 송신자의 개인키로 메시지를 암호화할 수 있으며, 수신자는 송신자의 공개키로 서명을 복호화함으로써 송신자의 신원을 확인할 수 있다.

사용자가 자원에 대한 특권(보기, 변경 등)을 갖는지의 여부가 인증에 의하여 결정된다. 예컨대, 시스템 관리자는 어느 사용자가 시스템에 액세스할 수 있는지 결정할 수 있으며, 각 사용자가 시스템 내에서 어떠한 권한을 갖는지, 즉 소정 파일에 대한 액세스 및 저장 공간의 용량 등을 결정할 수 있다. 권한 부여는 인증 이후에 수행되는 것이 통상적이다. 즉, 사용자가 객체로의 액세스를 요구하면, 시스템은 우선 사용자의 신원을 확인 또는 인증하고, 그 다음 그 사용자가 객체에 대한 액세스 권한을 갖는지 여부와, 사용자가 상기 객체를 어떻게 사용할 것인지 여부를 결정한다.

객체 보안화를 위하여 암호화도 사용될 수 있다. 메시지의 평문은 암호화에 의하여 암호문으로 변환된다. 암호화된 객체를 번역하기 위하여 수신자는 정확한 복호키를 획득하여야 한다(예컨대, 전술한 공개키 기반 구조 및 공개키 암호화에 대한 설명을 참조하기 바람). 종종, 객체를 암호화하는데 사용되는 암호기가 해킹(break)될 수 있지만, 암호화가 복잡할수록 복호키 없이 암호기를 해킹하기가 더욱 어려워진다. 강력한 암호화 시스템은 모든 가능키의 시도에 의해서도 암호기를 해킹할 수 없을 정도의 광범위한 가능키를 갖는다. 또한, 강력한 암호화 시스템은 기지의 코드 해킹 방법의 영향을 받지 않고, 모든 표준적인 통계 테스트에 랜덤하게 보일 것이다.

전체 컴퓨터 시스템을 보안화하기 위한 다른 종류의 보안법이 컴퓨터 위치에서 사용될 수 있다. 예컨대, 많은 기업은 기업 데이터 또는 프로그램에 대한 불법 사용자의 액세스를 방지하기 위하여 방화벽을 설치한다. 그러나, 방화벽은 손상될 수 있으며, 침입에 대한 컴퓨터 시스템의 안전성을 보장하지 않는다. 다른 문제점은 방화벽이 방화벽 뒤에 위치한 부정 사용자에 의하여 시스템이나 시스템 자원이 손상되지 않도록 보안화하지 않는다는 것이다.

메시지의 전송은 기밀화될 수 있다. 통상적으로, 서버와 클라이언트 사이에서의 암호화 통신을 제공하는 데에 전송 계층 보안(TLS: Transport Layer Security) 및 보안 소켓 계층(SSL: Secure Sockets Layer) 프로토콜이 사용된다. 이들 2개의 프로토콜은 대부분의 웹브라우저와 서버에 통합되어 있다.

감사 추적은 책임추적성(accountability)의 수행에 의하여, 즉 객체와 관련된 사용자의 행위(객체 요구 등)나 객체에 실제적으로 수행된 사용자의 행위(보기, 편집, 인쇄 등)를 추적함으로써 보안화를 제공한다. 감사 추적은 불법 변경으로부터 안전하여야 하며, 예컨대 불법 사용자가 감사 로그(audit log)로부터 자신들의 행위에 대한 증거를 제거하는 것이 허용될 수 없다. 요구와 행위에 대한 감사는 다량의 정보를 생성하기 때문에 감사 추적을 생성하는 시스템은 상기 정보를 저장하고 효율적으로 처리할 수 있는 능력을 갖추어야 한다.

전술한 보안 장치는 개별적으로 사용될 수 있고, 몇 가지가 복합적으로 사용되는 것이 더욱 일반적이다. 이러한 일반적인 장치 이외에도, 종래에 다른 보안화 방법들이 존재한다.

인터트러스트 테크놀로지스사(InterTrust Technologies Corporation)는 디지털 콘텐츠 권리 관리(digital rights management)에 대하여 수개의 특허를 획득하였다. 인터트러스트사의 디지박스(Digibox) 컨테이너 기술에 의하여 정보(콘텐츠 및 콘텐츠에 대한 액세스 관련 규칙을 포함함)를 암호화하고, 디지박스 컨테이너, 본질적으로는 소프트웨어 컨테이너에 저장할 수 있다. 상기 암호키와 컨테이너는 가상 분배 환경(VDE: Virtual Distribution Environment)에서 노드(node)로부터 노드로 전달된다. 상기 가상 분배 환경(VDE)은 전용의 하드웨어 또는 소프트웨어를 포함하거나 이들의 결합을 포함한다. 적절한 인터트러스트사의 소프트웨어를 구동시키는 VDE 내에 통합된 장치만이 상기 컨테이너의 정보를 볼 수 있다. 상기 VDE 내에서 감사 추적이 생성되고 저장되며 보여질 수 있다.

정당한 요구자가 보안화된 정보에 액세스하기 위하여 자신의 컴퓨터상에 특정 소프트웨어를 구동시키지 않고서도, 인터넷에서 얻을 수 있는 객체(코드, 문서, 이미지 및 소프트웨어 프로그램을 포함하며, 기본적으로는 디지털 형식으로 표현된 모든 것)를 보안화하는 발명이 요구되며, 책임추적성을 보장하는 안전한 감사 추적과 비반박성(non-refutability)도 요망된다.

객체 서버에 대한 안전성 제공의 처리 부담 및 하드웨어 부담(방대한 양의 감사 추적을 저장하기에 충분한 메모리를 포함함)을 경감시키기 위하여 감사 추적의 저장을 포함한 보안 의무를 제3자에게 넘기는 것이 바람직하다. 마지막으로, 광범위한 보안을 제공하고 감사 추적의 무결성(integrity) 및 비반박성을 증명하기 위하여 요구, 인증, 권한부여, 요구된 객체의 일련화, 요구된 객체의 넌스(nonce), 요구된 객체의 보안 정책 및 감사 추적에서 보안화된 객체의 기술(description)과 같은 정보를 저장하는 것이 바람직하다.

정당한 요구자가 보안화된 정보에 액세스하기 위하여 자신의 컴퓨터상에 특정 소프트웨어를 구동시키지 않고서도, 인터넷에서 얻을 수 있는 객체(코드, 문서, 이미지 및 소프트웨어 프로그램을 포함하며, 기본적으로는 디지털 형식으로 표현된 모든 것)를 보안화하는 발명이 요구된다. 예컨대, 학생들은 제한된 예산으로 인하여, 자신의 컴퓨터를 가지고 있는 경우에도, 강좌 노트 및 시간표와 같은 정보(학교는 점점 인터넷을 통하여 정당한 사용자에게 이러한 정보의 사용을 허용하고 있음)의 다운로드를 가능하게 해주는 소프트웨어의 구입이 당연히 기대될 수 있는 것은 아니다. 디지털 콘텐츠 권리 관리 시스템의 바람직한 추가적 특징은 암호키를 암호화된 데이터와 함께 전달하기보다 요구자와 '보안 서버' 사이에서 안전하게 전달되는 1회용 암호키를 제공하고 안전성을 제공하는 처리상의 부담을 객체 서버로부터 경감시키기 위하여 대부분의 보안화 '의무'를 제3자에게 넘기는 것을 포함한다. 객체가 요구자에게 전송된 이후에도 디지털 콘텐츠 권리 관리 시스템이 객체에 대한 보안을 제공하는 것도 바람직하다.

발명의 상세한 설명

본 발명은 네트워크를 통하여 분배되는 객체(코드, 문서, 이미지, 소프트웨어 프로그램 등 디지털 형식으로 표현된 모든 것)를 보안화하기 위한 방법과 시스템을 제공한다. 보안화란 특정 수신자들이 객체에 대하여 특정 행위(즉, 보기, 인쇄, 편집, 복사)를 하지 못하도록 제한하는 것이다.

보안화된 객체 및 비보안화된 객체 전부를 포함한 객체 서버에는 객체가 보안화되어야 하는지 여부와, 보안 정책(객체가 받는 보안화의 종류와 등급)은 무엇인지를 규정한 소프트웨어가 구비된다. 상기 보안 정책에는 객체가 인쇄될 수 있는지, 편집될 수 있는지 등의 행위와 관련된 행위 정책뿐만 아니라, 객체를 볼 수 있는 사람, 객체의 수명, 객체가 보여지는 횟수에 대한 제한 사항이 포함된다. 객체 제어는 보안 정책을 실행시키는 메커니즘이다.

객체 서버가 객체에 대한 요구를 수신하면, 소프트웨어는 상기 요구된 객체가 보안화되었는지 여부를 검사한다. 객체가 보안화되지 않았으면, 서버는 상기 객체를 요구자에게로 전송할 것이다. 객체가 보안화되었을 경우, 소프트웨어는 요구된 객체의 일련화, 넌스, 보안 정책 및 기술(description)뿐만 아니라, 본래 요구의 시간 및 인증을 포함하는 새로운 객체를 생성한다. 상기 새로운 객체는 요구 브라우저로하여금 '보안 서버'를 지정하도록 하는 재지정 명령과 함께 요구 브라우저에 회답으로 반송된다.

보안 서비스를 제공하는 소프트웨어에 구비된 보안 서버는 재지정된 요구를 수신하고 인증한 다음, 보안 전송을 통하여 객체를 포함한 서버나 자신의 고유 캐시로부터 상기 요구된 객체를 획득한다. 그 다음, 보안 서버는 강력하고 비적응적 암호화를 이용하여 상기 요구된 객체를 암호화하고, 이동 코드(수신자에 의한 뚜렷한 설치나 실행 없이 원격지 시스템으로부터 송신되고, 네트워크를 통하여 전송되며, 로컬 시스템에 다운로드되고 실행하는 소프트웨어), 보안 정책 및 객체 제어와 결합시킨다. 그 결과로서의 패키지가 상기 재지정된 요구에 대한 응답으로 요구 컴퓨터에 반송된다.

상기 요구 컴퓨터는 요구된 객체를 제공하기 위하여 상기 이동 코드의 실행을 시도한다. 상기 이동 코드는 객체 제어의 실체화(instantiation)를 보증하기 위한 테스트를 실행시키고, 이들 제어가 적절하게 실체화된 경우 요구자는 요구의 인증이 만족되었을 때 보안 전송을 통하여 요구자에게 전송되는 복호키를 요구한다. 상기 복호키는 당해 특정 객체를 복호화하는 데에만 사용되는 1회용 키이다. 상기 이동 코드가 성공적으로 실행되어 복호키가 획득되면, 상기 요구된 객체는 보안 정책과 객체 제어의 제약을 받는다.

보안 서버와 관련된 소정의 행위와 객체에 관한 요구자의 행위의 기술자(descriptor)가 보안 서버의 시스템 관리자와 콘텐츠 소유주와 같이 허가된 개인에 의한 검토에 사용될 수 있는 로그 파일에 기록된다. 상기 로그 파일을 사용하여 객체에 액세스된 시간과 객체에 액세스된 횟수 등과 같은 획득된 정보뿐만 아니라, 누가 어느 객체를 요구하였는지, 객체가 전달되었는지, 각각의 객체에 대하여 어떠한 종류의 보안 정책이 적절한지, 요구자에 의하여 객체에 행해진 소정의 행위를 상술하는 감사 추적을 작성한다.

상기 보안 서버는 요구된 객체를 보안화하고 전달하는 것에 관련된 대부분의 행위를 실행시키는 데에 사용된다. 따라서, 객체 서버는 보안 이슈에 처리 자원을 소모하지 않는 대신에 정보에 대한 요구를 처리하는 데에 전용된다. 또한, 객체 서버에 대한 모든 설정 시각과 관리가 서버의 시스템 관리자에 의하여 다루어지기 때문에, 객체 서버의 소유자에게는 절감의 결과가 발생된다.

본 발명의 방법 및 시스템은 객체의 요구와 요구된 객체의 제공에 관계된 모든 컴퓨터에 공통의 소프트웨어를 설치할 필요가 없다는 점에서 다른 객체 보안 방법 및 시스템과 다르다. 또한, 객체를 암호화 및 복

호화하는 데에 사용되는 키는 1회용 키이고, 암호화된 객체와 함께 전달되지 않는다.

도면의 간단한 설명

도 1은 본 발명에 따른 객체 보안 시스템의 구성 요소에 대한 블럭도.

도 2a는 본 발명에 따라 객체가 보안화되는 방법을 도시한 흐름도.

도 2b는 본 발명에 따라 객체가 보안화되는 방법을 도시한 흐름도.

도 3a는 본 발명에 따라 보안화된 객체에 대한 요구자의 행위의 로그 파일이 생성되는 방법을 도시한 흐름도.

도 3b는 본 발명에 따라 객체 서버 행위의 로그 파일이 생성되는 방법을 도시한 흐름도.

실시예

도 1을 참조하면, 요구자 장치(10)(본 실시예에서는 컴퓨터이지만, 클라이언트/서버 관계에서 클라이언트로 동작할 수 있는 것을 포함하는 장치임)와, 보안화되어야할 객체를 표시하는 보안 소프트웨어(14)와 객체(16)를 포함한 객체 서버(12)와, 보안 서비스를 제공하기위한 소프트웨어(94)를 포함한 보안 서버(18)가 전부 네트워크[본 실시예에서는 인터넷(20)]에 접속된다. 객체(16)는 코드, 문서, 이미지, 소프트웨어 프로그램 등과 같이, 디지털 형식으로 표현된 것을 포함한다. 보안화된 객체에 대한 불법 액세스를 획득하는 데에 사용되는 컴퓨터나 기록기와 같은 장치 또는 사람에 해당하는 공격자(22)가 존재한다. 여기에서는 1개의 요구자 장치(10), 객체 서버(12) 및 보안 서버(18)가 설명되지만, 본 발명의 방법 및 시스템은 복수의 요구자 장치(10), 객체 서버(12) 및 보안 서버(18)를 수용하는 것이 가능하다.

본 실시예에서, 상기 객체 서버(12)와 보안 서버(18)는 하이퍼텍스트 전송 프로토콜(http) 서버이다. 상기 요구자 장치(10)는 월드와이드 웹 브라우저(24)로 동작하는 소프트웨어 프로그램을 구동시켜야 한다. 상기 요구자 장치(10)로부터의 객체(16)에 대한 요구가 브라우저(24)에 의해 http 요구를 통하여 객체 서버(12)로 중계된다. 마찬가지로, 요구에 대한 응답도 http 프로토콜을 따른다.

전술한 바와 같이, 객체 서버(12)는 보안 소프트웨어(14)를 구동시키며, 본 실시예에서 상기 보안 소프트웨어(140)는 http 서버 소프트웨어의 확장이다. 인가된 시스템 관리자는 상기 보안 소프트웨어(14)를 사용하여 어느 객체(16)가 보안화되지 않았는지와, 어느 것이 보안화될 것인지 지정한다. 객체(16)가 보안화된 것으로 지정되면, 상기 보안 소프트웨어(14)는 관리자로하여금 객체(16)에 대한 보안화 종류와 등급(즉, 보안 정책)을 지정하게 한다. 상기 보안 정책에는 객체가 인쇄되는지, 편집되는지 등에 관련된 행위 정책뿐만 아니라 객체를 보는 사람과, 객체의 수명(즉, 시간적인 제한 사항)과, 객체가 보여지는 횟수(즉, 수적 제한)에 대한 제한 사항이 포함된다. 요구자가 객체에 행하는 행위들은 요구자의 신원에 따라 다르다. 객체 제어는 보안 정책을 실행하는 메커니즘이다.

보안 서버(18)도 http 서버 소프트웨어의 확장인 소프트웨어(94)를 구동시킨다. 이 소프트웨어(94)는 객체에 보안 서비스를 제공한다.

도 2a에 있어서, 요구자는 객체를 요구한다(단계 26). 상기 요구된 객체를 저장하는 객체 서버가 상기 요구를 수신한다(단계 28). 상기 객체 서버가 독립적인 인증 정책을 가지고 있을 경우, 객체 서버는 그 정책을 수행할 것이고, 요구 수신시 그를 인증할 것이다. 상기 보안 소프트웨어는 http 요구를 검사하여 그 요구가 보안화된 객체에 대한 것인지 결정한다(단계 30). 요구된 객체가 보안화되지 않았을 경우, 요구된 객체는 요구자에게 전송된다(단계 32).

그러나, 객체가 보안화된 경우(단계 30), 상기 보안 소프트웨어는 요구에 대한 응답 내에 포함되는 보완된 요구를 생성하고(단계 34), 이어서 보안 서버로 재전송된다. 상기 보완된 요구는 요구된 객체의 기술(description), 보안 정책, 넌스(nonce) 및 일련화(하나의 승인된 버전의 객체만이 사용 가능함을 보증함)뿐만아니라 본래 요구의 시간 및 인증을 포함하는 암호화 데이터가 포함된 객체이다. 인증에 관한 정보는 객체 서버가 독립적인 인증 정책을 가지고 있는지의 여부에 따라 좌우된다. 인증 정책이 존재하는 경우, 상기 보완된 요구는 인증의 결과를 포함한다. 인증 정책이 존재하지 않는 경우, 그 정보가 상기 보완된 요구에 포함된다.

암호화에 의하여 다양한 서비스가 제공된다. 상기 암호화는 요구에 대한 인증 및 권한 부여를 지원할뿐만 아니라, 파일의 무결성을 보호할 수 있다(즉, 불법 변경의 방지). 여기에서는 암호화를 사용하여 요구자의 개인 정보(privacy)도 보호할 수 있다. 암호화에 대한 다른 기능으로는 부인 방지(non-repudiation)와 변경의 검출이 있다. 강력하고 비적응적인 암호화를 지원하는 프로토콜이 사용된다. 프로토콜은 사용된 암호화의 종류를 결정하고, 암호화가 이루어지기 전에 요구자와 보안 서버간 교환이 필요한지 여부가 결정된다(예컨대, 키는 수신자가 서버에서 암호화된 객체를 복호화할 수 있도록 교환될 필요성이 많다).

상기 보완된 요구는 보안 서버로의 요구의 재전송 명령과 함께 요구자에 대한 응답에 포함된다. 상기 재전송은 요구자에 대하여 투명하여야 한다.

상기 보안 서버 소프트웨어는 상기 보완된 요구를 복호화한다(단계 38). 상기 보완된 요구를 암호화 및 복호화하기 위한 공유키는 객체 서버와 보안 서버에 존재한다. 이 키는 상기 소프트웨어가 객체 서버상에 설치될 때 생성된다.

다음으로, 상기 보안 서버 소프트웨어는 상기 보완된 요구가 잘 구성된 요구에 대한 조건을 만족하는지 검사한다(단계 40). 잘 구성된 요구에 대한 조건이 만족되지 않을 경우, 보안 서버가 객체 서버에 무효 요구를 표시하는 메시지를 반송한다(단계 42). 상기 객체 서버는 요구자에게 무효 요구에 관한 메시지를 전송한다. 상기 객체 서버에 대한 시스템 관리자는 상기 메시지의 전송 여부를 결정한다.

상기 요구가 유효한 경우, 보안 서버 소프트웨어는 상기 요구를 인증한다(단계 44). 보안 서버 소프트웨어

10-4

는 상기 재전송된 요구 표제(heading)에서의 시간과 인증을 보완된 요구에 포함된 것들과 비교할 것이다. 상기 보안 서버 소프트웨어가 요구를 인증할 수 없으면(예컨대, 2개의 요구 시간이 상이하여 재전송 공격이 표시되거나, 재전송된 요구에서의 요구자 신원이 보완된 요구에서의 요구자 신원과 다를 경우), 인증이 만족되지 않았음을 표시하는 메시지가 객체 서버로 반송된다(단계 46). 요구가 인증되면, 보안 서버 소프트웨어는 상기 요구를 복호화하고, 보안 서버의 캐시 또는 객체 서버로부터 요구된 객체를 획득한다(단계 48). 상기 보안 소프트웨어는 요구가 있으면 상기 객체를 보안 서버로 전달할 것이다. 보안 서버가 객체 서버로부터 객체를 획득하여야 할 경우, 객체는 보안 전송을 통하여 전달된다.

일단 보안 서버가 요구된 객체를 갖게되면, 보안 서버 소프트웨어는 강력한 암호화 및 비적응적 암호화를 사용하여 상기 객체를 암호화하고, 상기 객체를 이동 코드(수신자에 의한 뚜렷한 설치나 실행 없이 원격지 시스템으로부터 송신되고, 네트워크를 통하여 전송되며, 로컬 시스템에 다운로드되고 실행하는 소프트웨어), 보완된 요구 내에 포함된 인증을 갖는 보안 정책 및 객체 제어와 결합시킨다(단계 50). 보안화된 요구 객체의 암호화는 변경 방지 및 부인 방지(즉, 트랜잭션에의 당사자가 당해 트랜잭션과의 관계를 부정하게 부인할 수 없는 것) 도구로서뿐만 아니라, 무결성, 개인 정보 및 인증(적절한 경우)을 보증함으로써 객체와, 객체의 요구자 및 제공자에 대한 보호 역할을 수행한다. 이어서, 결과 패키지가 요구자에게 전송된다(단계 52; 도 2b의 단계 B 참조).

도 2b에 있어서, 요구자는 응답을 수신하고 상기 이동 코드의 실행을 시도한다(단계 54). 상기 이동 코드가 실행되면, 요구된 객체에 대한 보안 정책과 객체 제어가 요구자의 컴퓨터상에서 실체화된다(단계 54). 상기 이동 코드는 객체 제어가 정확하게 실체화되었는지를 결정하기 위한 테스트를 실행한다. 정확하게 실체화된 경우, 요구자가 복호키를 필요로 한다면(단계 56) 요구자는 보안 서버에 복호키를 요구한다(단계 58). 보안 서버 소프트웨어는 상기 요구를 인증한다(단계 60). 상기 보안 서버 소프트웨어가 당해 요구를 인증할 수 없다면, 그 결과에 대한 메시지가 객체 서버로 전송된다(단계 62). 그러나, 메시지가 인증된 경우, 상기 보안 서버 소프트웨어는 요구된 키를 보안 전송에 의하여 요구자에게 반송시키고(단계 64), 요구된 객체는 복호화된다(단계 66). 보안 서버가 객체를 암호화 및 복호화하는 데에 사용한 키는 1회용 키이다. 상기 1회용 키는 보안 서버 소프트웨어의 설치시 결정되는 키를 랜덤하게 생성하기 위한 '시드(seed)'에 의해 제공되거나, 종래에 공지된 기타의 수단, 가장 일반적으로는 인증서에 의하여 제공된다.

일단 상기 이동 코드가 실행되면, 요구자는 보안 정책이나 객체 제어에 의해 객체상에 부과된 제약을 받는 객체를 볼 수 있다(단계 68).

도 3a에 도시된 바와 같이, 요구자에 의하여 객체에 수행된 행위의 로그 파일은 감사 추적의 설정을 위하여 유지된다. 상기 로그 파일은 보안 서버 시스템 관리자에 의한 검토용으로 사용될 수 있다. 상기 로그 파일을 사용하여 누가 어떠한 객체를 요구하였는지, 객체가 전달되었는지, 이들 객체 각각에 대하여 어떠한 종류의 보안 정책이 적합한지를 상술하는 감사 추적을 작성한다.

요구자가 객체와 관련된 행위(객체 보기, 인쇄, 편집 등)를 시도하면, 객체 제어는 네트워크에 대해 설정된 접속이 존재하는지 여부를 결정할 것이다(단계 82). 개방형 접속이 존재하는 경우, 암호화된 행위의 기술자(descriptor)가 보안 서버에 전송될 것이고, 보안 서버는 일부 다른 데이터와 함께 상기 기술자를 로그 파일에 기록할 것이다(단계 88). 로그 파일에 기록되는 기타의 자료에는 '로컬 데이터', 즉 서버의 현지 시간과 신원, 시각 및 요구자 네트워크의 IP 주소를 포함한 서버측의 데이터가 포함된다. 일단 상기 정보가 보안 서버로 전송되고, 요구자에게 검증이 전송되면(단계 94) 객체에 대한 행위가 허용된다(단계 90). 예컨대, 전술한 바와 같이, 요구자는 상기 이동 코드가 성공적으로 실체화되고, 보안 서버로부터 복호키가 수신된 경우에만 요구된 객체를 볼 수 있다. 우선, 요구자의 컴퓨터에 객체가 표시되면, 상기 이벤트, 즉 객체 보기에 대한 기술자가 보안 서버로 전송된다. 요구자에게 어떠한 검증도 전송되지 않은 경우, 객체상에 행위를 수행하는 요구자의 요구가 거부된다(단계 92).

보안 서버에 대하여 안전하게 설정된 접속이 존재하지 않으면, 객체 제어는 그러한 접속을 보안 서버에 대하여 설정하고자 한다(단계 84). 상기 접속이 설정되면(단계 86), 암호화된 행위의 기술자가 보안 서버로 전송될 것이고, 상기 보안 서버는 상기 기술자와 전술한 기타의 데이터를 로그 파일에 기록할 것이다(단계 88). 이어서, 객체에 대한 행위가 허용된다(단계 90). 그러나 접속이 설정될 수 없으면(단계 86), 객체상에 행위를 수행하는 요구자의 요구가 거부된다(단계 92).

도 3b에 도시된 바와 같이, 보안 서버는 보안화된 객체에 대해 수행된 행위의 기술자들을 로그 파일에 저장한다. 이들 행위에는 객체 요구에 대한 응답, 요구자에게로의 객체 전송, 복호키에 대한 요구 수신 및 요구자에게로의 복호키 전송이 포함된다. 보안 서버가 행위를 수행할 때(단계 74), 시스템 소프트웨어는 당해 행위가 보안화된 객체의 전송에 관련된 것인지, 또는 복호키에 대한 요구에 관련된 것인지 결정한다(단계 76). 상기 행위가 보안화된 객체의 전송이나 복호키에 대한 요구에 관련되지 않은 경우, 로그 파일에 아무것도 기록되지 않는다(단계 80). 그러나, 상기 행위가 보안화된 객체 또는 복호키에 관련되어 있는 경우, 시간, 로컬 데이터 및 요구자의 네트워크 IP 주소와 함께 행위의 기술자가 로그 파일에 기록된다(단계 78). 예컨대, 보안 서버가 보안화된 객체에 대하여 보완된 요구를 수신하면, 상기 보안 서버는 상기 로그 파일에 보완된 요구를 저장하고, 상기 보완된 요구와 함께 적어도 시간, 로컬 데이터 및 요구자의 네트워크 IP 주소가 저장된다. 보안 서버가 이동 코드와 결합된 객체를 포함하는 패키지를 요구자에게 전송하였을 때, 이 행위의 기록이 상기 로그 파일에 기록된다.

다른 실시예에 있어서, 요구자는 객체상에 행위를 취할 수 있지만 '무선화(untethered)'(즉, 보안 서버에 접속되지 않은 상태) 된다. 보안 정책에 의하여 무선상의 행위가 허용될 경우, 요구자의 행위는 요구자의 장치에 기록되고, 요구자가 보안 서버에 대한 접속을 설정하였을 때 보안 서버로 전송된다. 설정된 시간 프레임 내에 네트워크로의 접속이 구축되지 않으면 객체에 대한 접근이 제한되도록 제어가 이루어진다.

또 다른 실시예에 있어서, 보안 서버의 행위에 대한 기술자들은 로그 파일에 기록되기 전에 암호화될 수 있다. 이 실시예는 시스템 관리자 이외의 사람에게 상기 로그 파일에 대한 액세스가 허용되었을 때 사용될 수 있다.

10-5

(57) 청구의 범위

청구항 1

통신 네트워크에 있어서, 네트워크 내에 분산되는 객체에 수행된 행위와 요구된 행위의 로그 파일을 제공함으로써 객체를 보안화하는 보안화 시스템으로서, 상기 보안화 시스템은

a) 객체를 포함하고, 보안화될 객체와 보안화된 객체의 보안 정책을 규정한 소프트웨어 프로그램을 구동시키며, 네트워크에 접속되는 객체 서버와,

b) 네크워크에 접속되고, 상기 객체 서버에 대하여 객체를 요구하는 요구자 장치와,

c) 네트워크에 접속되고, 상기 소프트웨어 프로그램에 의하여 보안화된 것으로 규정된 객체에 대하여 보안 서비스를 제공하는 다른 소프트웨어 프로그램을 구동시키는 보안 서버를 포함하고, 상기 보안 서비스를 제공하는 소프트웨어는

i) 상기 요구자 장치로부터 요구된 객체에 대하여 재지정 및 보완된 요구를 수신하는 수단으로서, 상기 보완된 요구는 상기 요구자 장치의 최초 요구에 대응되고, 상기 객체 서버에 의하여 생성되며, 상기 보완된 요구는 요구된 객체의 기술(description), 보안 정책, 넌스(nonce) 및 일련화(serialization)뿐만아니라 최초 요구의 시간 및 인증과 관련된 암호화 데이터를 포함하는 객체인 것인 요구된 객체에 대한 재지정 및 보완된 요구의 수신 수단과,

ii) 상기 요구된 보안화 객체가 저장되는 객체 서버로부터 또는 캐시로부터 상기 요구된 보안화 객체를 획득하는 수단과,

iii) 상기 요구된 보안화 객체를 암호화하는 수단과,

iv) 상기 요구된 보안화 객체를 이동 코드와, 보안 정책 및 객체 제어와 결합시키는 수단과,

v) 결과 파일을 요구자 장치에 전송하는 수단으로서, 상기 요구자 장치는 상기 이동 코드를 실행시켜 상기 요구자 장치에 요구된 객체를 제공하여야 하며, 상기 요구자 장치의 사용자는 객체를 사용하고 보기 위하여 상기 이동 코드의 실행시 요구자 장치상에 위치되는 객체 제어와 보안 정책에 의존되는 것인 결과 파일의 전송 수단과,

vi) 상기 객체 제어의 적절한 실체화 여부를 확인하는 수단과,

vii) 상기 키 요구의 인증 만족에 따라 복호키를 요구자 자이에 제공하는 수단과,

viii) 보안 서버에 저장되는 것으로서, 이벤트에 대한 정보를 로그 파일에 기록하는 수단을 포함하고, 상기 이벤트는

A) 요구자 장치에 의해 개시된 요구 보안화 객체에 대한 행위 요구와,

B) 요구자 장치에서 상기 요구 보안화 객체상에 행해진 행위와,

C) 상기 보안 서버에 의해 행해지는 것으로서, 요구 보안화 객체의 보안화와 관련된 행위

를 포함하는 그룹에 속하는 것인 보안화 시스템.

청구항 2

제1항에 있어서, 상기 로그 파일은 감사 추적을 생성하는 데 사용되는 것인 보안화 시스템.

청구항 3

제1항에 있어서, 상기 기록된 정보는 이벤트의 시각인 것인 보안화 시스템.

청구항 4

제1항에 있어서, 상기 기록된 정보는 로컬 데이터인 것인 보안화 시스템.

청구항 5

제1항에 있어서, 상기 기록된 정보는 이벤트를 개시한 요구자 장치의 네트워크 IP 주소인 것인 보안화 시스템.

청구항 6

제1항에 있어서, 상기 로그 파일에 기록된 정보는 이벤트의 기술자(descriptor)를 포함하는 것인 보안화 시스템.

청구항 7

제1항에 있어서, 상기 로그 파일에 기록된 정보는 보안 서버에 전송되는 요구를 포함하는 것인 보안화 시스템.

청구항 8

제1항에 있어서, 상기 요구자 장치에 의하여 상기 보안 서버로 전송되는 정보는 프로토콜에 의하여 암호화되는 것인 보안화 시스템.

청구항 9

제8항에 있어서, 정보에 대한 암호화를 포함하는 프로토콜은 강력한 암호화를 제공하는 것인 보안화 시스템.

청구항 10

제8항에 있어서, 정보에 대한 암호화를 포함하는 프로토콜은 비적응적 암호화(non-malleable encryption)를 제공하는 것인 보안화 시스템.

청구항 11

제1항에 있어서, 요구자 장치에서 개시된 행위 요구에 대한 정보를 기록하기 위하여 요구자 장치와 보안 서버간의 접속을 설정하는 수단을 더 포함하고, 상기 접속은 상기 요구자 장치와 상기 보안 서버간에 접속이 현존하지 않는 경우에 설정되는 것인 보안화 시스템.

청구항 12

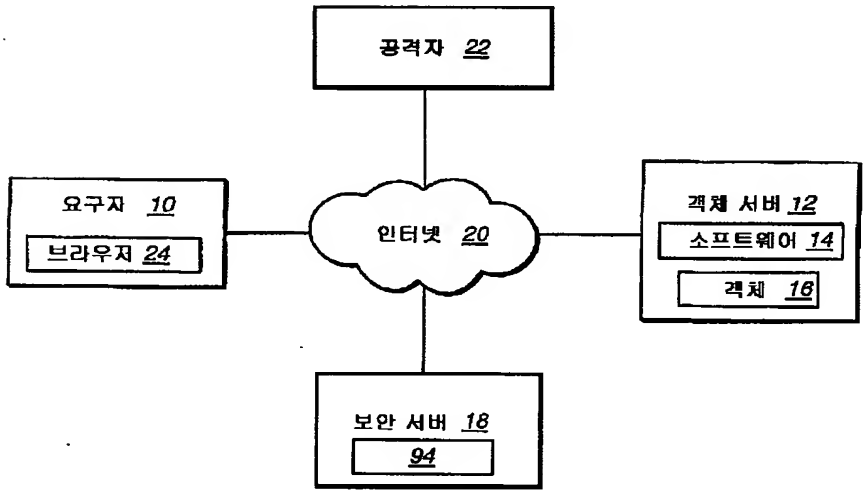
제11항에 있어서, 상기 요구자 장치와 상기 보안 서버 사이에 접속이 설정될 수 없을 때, 보안화된 객체에 대하여 요구된 행위를 거부하는 수단을 더 포함하는 것인 보안화 시스템.

청구항 13

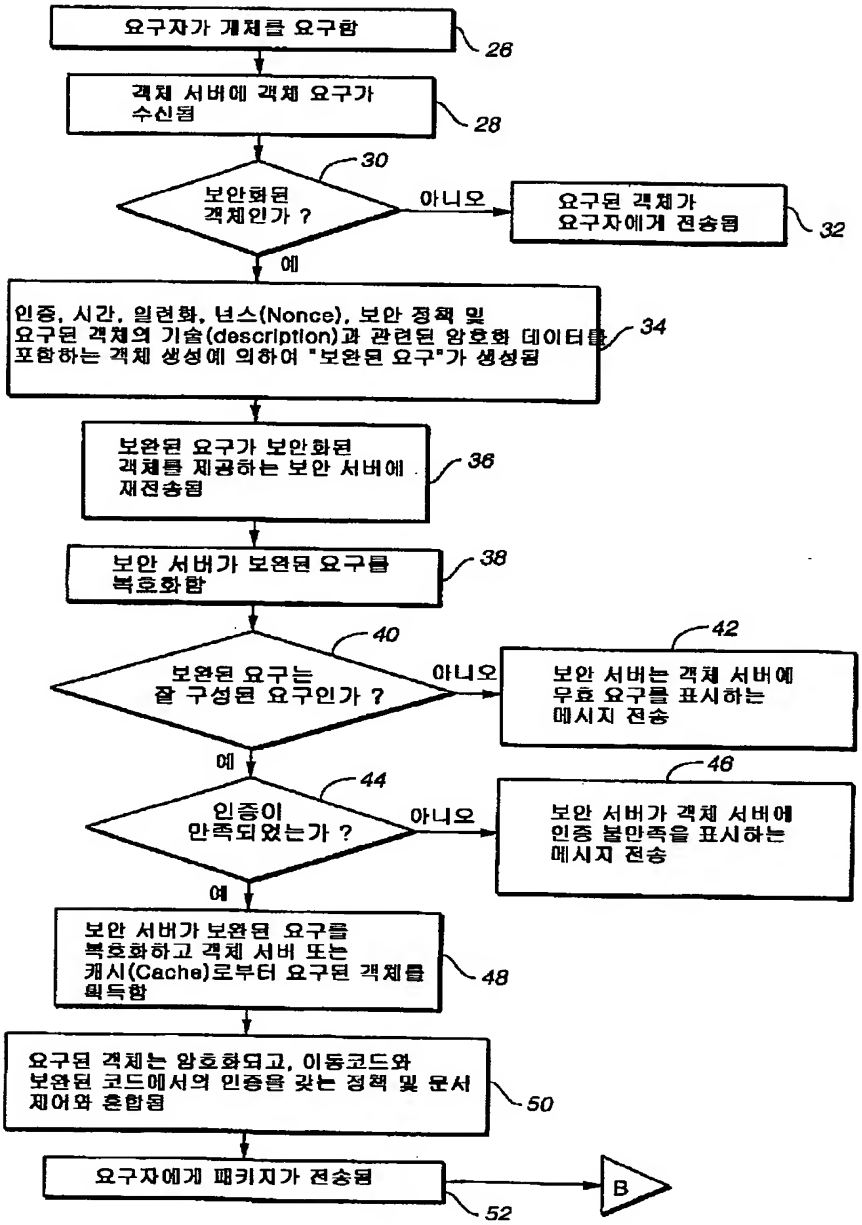
제1항에 있어서, 상기 요구자 장치와 상기 보안 서버 사이에 접속이 설정되었을 때, 무선의 요구자 장치가 요구자 장치상의 파일에 요구된 보안화 객체에 대한 행위를 기록하고, 상기 파일을 보안 서버로 전송하는 수단을 더 포함하는 것인 보안화 시스템.

도면

도면1

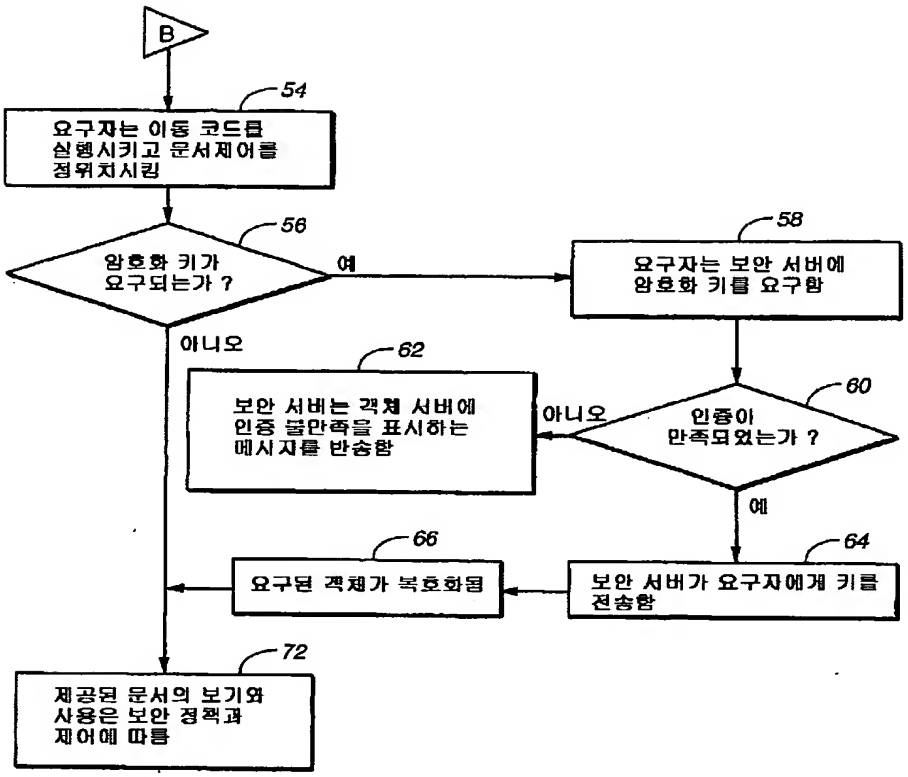


도면2a


요구자가 개체를 요구함
26
객체 서버에 객체 요구가 수신됨
28
30
보안화된 객체인가 ?
아니오
요구된 객체가 요구자에게 전송됨
32
예
인증, 시간, 일련화, 넌스(Nonce), 보안 정책 및 요구된 객체의 기술(description)과 관련된 암호화 데이터를 포함하는 객체 생성에 의하여 "보완된 요구"가 생성됨
34
보완된 요구가 보안화된 객체를 제공하는 보안 서버에 재전송됨
36
보안 서버가 보완된 요구를 복호화함
38
40
보완된 요구는 잘 구성된 요구인가 ?
아니오
42
보안 서버는 객체 서버에 무효 요구를 표시하는 메시지 전송
예
44
인증이 만족되었는가 ?
아니오
46
보안 서버가 객체 서버에 인증 불만족을 표시하는 메시지 전송
예
보안 서버가 보완된 요구를 복호화하고 객체 서버 또는 캐시(Cache)로부터 요구된 객체를 획득함
48
요구된 객체는 암호화되고, 이동코드와 보완된 코드에서의 인증을 갖는 정책 및 문서 제어와 혼합됨
50
요구자에게 패키지가 전송됨
52
B

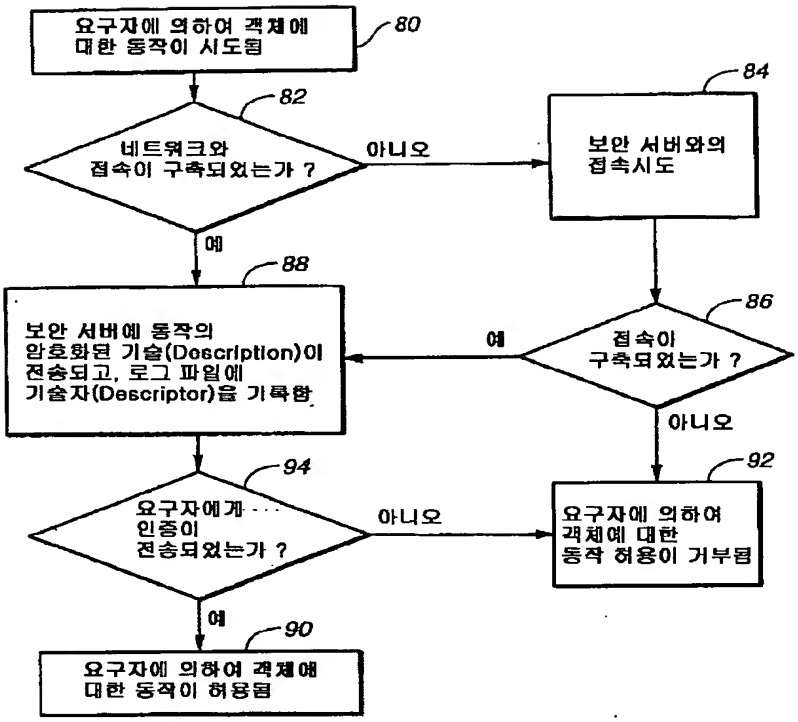
10-8

도면2b

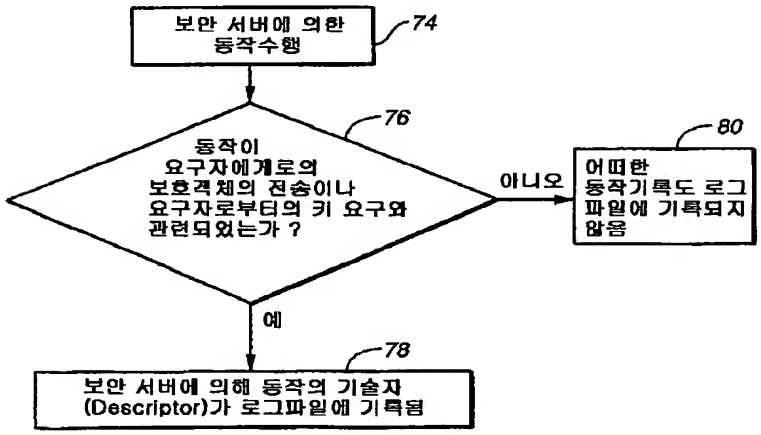

B
54
요구자는 이동 코드를 실행시키고 문서제어를 정위치시킴
56
암호화 키가 요구되는가 ?
예
58
요구자는 보안 서버에 암호화 키를 요구함
아니오
62
보안 서버는 객체 서버에 인증 불만족을 표시하는 메시지를 반송함
아니오
60
인증이 만족되었는가 ?
예
66
요구된 객체가 복호화됨
64
보안 서버가 요구자에게 키를 전송함
72
제공된 문서의 보기와 사용은 보안 정책과 제어에 따름

10-9

도면3a



도면3b



10-10